

# Package ‘session’

January 2, 2012

**Description** Utility functions for interacting with R processes from external programs. This package includes functions to save and restore session information (including loaded packages, and attached data objects), as well as functions to evaluate strings containing R commands and return the printed results or an execution transcript.

**Title** Functions for interacting with, saving and restoring R sessions.

**Version** 1.0.3

**Date** 2011-08-26

**Maintainer** Gregory R. Warnes <greg@warnes.net>

**Author** Gregory R. Warnes.

**License** GPL (>= 2)

**Repository** CRAN

**Date/Publication** 2011-08-27 05:09:02

## R topics documented:

save.session . . . . .	1
texteval . . . . .	3

<b>Index</b>	<b>5</b>
--------------	----------

---

save.session	<i>Save and restore session information, including loaded packages and attached data objects.</i>
--------------	---

---

## Description

Save and restore session information, including loaded packages and attached data objects.

**Usage**

```
save.session(file=".RSession", ...)  
restore.session(file=".RSession", ...)
```

**Arguments**

file	Filename for the session information.
...	Optional arguments for save() or load().

**Details**

These two functions save and restore R session information. In addition to the objects in the session, the list of currently loaded packages and the search path are (re)stored.

Open graphics devices, sinkss, pipes, etc. will not be stored. `save.session` issues a warning to this effect if any graphics devices are open.

**Value**

No return value.

**Note**

Future enhancements may allow the restoration of the size, location, and graphics settings of graphics devices.

**Author(s)**

Gregory R. Warnes <greg@warnes.net>

**See Also**

[save](#), [save.image](#), [load](#)

**Examples**

```
ls(all=TRUE) # show all data objects  
search() # list search path  
  
# save the current R session to the file "RSession.Rda"  
save.session("RSession.Rda")  
  
## Not run:  
# exit R without saving data  
q("no")  
  
# restart R  
R  
  
## End(Not run)
```

```
# load a saved R session from "RSession.Rda"
restore.session("RSession.Rda")

ls(all=TRUE) # show all data objects
search() # list search path
```

---

texteval	<i>Evaluate string(s) containing R commands and return the text transcript or printed results</i>
----------	---

---

### Description

Evaluate string(s) containing R commands and return the text transcript or printed results

### Usage

```
capture(expression, collapse=NULL)
texteval(sourceText, collapse=NULL, echo=TRUE)
printed(sourceText, collapse=NULL)
```

### Arguments

expression	R expression to be evaluated
sourceText	Vector of string to be evaluated.
collapse	Line separator. Defaults to NULL
echo	Should commands be shown in output. Defaults to TRUE

### Details

capture captures the results of executing expression using a textConnection. texteval and printed parse and evaluate the contents of sourceText using source and the results are captured using a textConnection. If collapse is NULL, a vector of strings is returned, one element for each line of output. (Empty strings for blank lines). If collapse is non-NULL, the a single character string is formed by pasting the individuals elements together separated by this value. When echo is TRUE, texteval will return a transcript that includes both printed output and evaluated commands. When echo is FALSE, texteval will return only the printed output. printed always returns only the printed output.

These functions were created to allow strings provided from external processes (for example by rpy or RSPerl) to be evaluated as if they were scripts.

### Value

A single character string if collapse is non-NULL, otherwise a vector of character strings.

**Author(s)**

Gregory R. Warnes <greg@warnes.net>

**See Also**

[source](#), [textConnection](#), [sink](#), [parse](#), [eval](#)

**Examples**

```
# define a script string
script <- "x <- rnorm(100)\ny <- x + rnorm(100,0.25)\nsummary(lm(y~x))"

# evaluate the script string, returning a transcript.
result <- texteval(script, "\n")
cat(result)

# evaluate the script string, returning the printed output.
result <- printed(script, "\n")
cat(result)
```

# Index

## \*Topic **data**

save.session, 1

texteval, 3

## \*Topic **programming**

save.session, 1

texteval, 3

capture (texteval), 3

eval, 4

load, 2

parse, 4

printed (texteval), 3

restore.session (save.session), 1

save, 2

save.image, 2

save.session, 1

sink, 4

source, 4

textConnection, 4

texteval, 3